Amendments To the Specification:

Please replace paragraph [0060] with the following amended paragraph:

[0060] Referring now to Fig. [[___]] 5, a method employed in one of the embodiments of the present invention uses a level 2 SATA port for host and device ports and a FIS FIFO between the host ports and device ports to avoid any data drop out. The level 2 SATA port responds immediately to HOLD/HOLDA rather than relaying the primitives and waiting for response from the other port. Fig. 5 shows a high-level block diagram of a switch 200 , switching within layer 2 and in accordance with an embodiment of the present invention. The switch 200 is shown to comprise a SATA level 2 host port 210, a SATA level 2 host port 220, a SATA level 2 device port 230, a FIS payload FIFO 245, a multiplexer 242a , a multiplexer 242b, a demultiplexer 243, an active host selection circuit 241, and a switch initialization circuit 244.

Please replace paragraph [0154] with the following amended paragraph:

[0154] It is obvious to one of ordinary [[skillin]] skill the art to extend embodiments of an SATA active switch of the present invention [[to SATA]] to an ATA Active Switch. Figs. 11a and 11b show such embodiments of SATA to ATA active switch that allow concurrent access by two hosts connected to a switch via a SATA link to a storage unit connected to a switch via an ATA link.

Please replace paragraph [0157] with the following amended paragraph:

[0157] Fig. [[11a]] 11b shows another embodiment of SATA to ATA switch 700 according to an embodiment of the present invention. The switch 700 is the same as the switch 500 of Fig.10a with the following differences:

- The SATA level 3 device port 530 in switch 500 is replaced with a SATA layer 3 to ATA Bridge 730
- The SATA link 531tx, 531rx in switch 500 are replaced with an ATA link 736.

Please replace paragraph [0158] with the following amended paragraph:

[0158] The SATA layer 3 to ATA Bridge 730 [[comprises a]] comprises a ATA Transport Layer 733, and a ATA Interface Bridge 732. The ATA Interface Bridges 732 is connected to the ATA link 736 and converts (bridges) the activity on the ATA bus 736 to the activity on the Transport layer interface 733io and visa versa. The Transport Layer 733 is the same as the Transport Layer 413 of Fig. [[9..]] 9.

Please replace paragraph [0163] with the following amended paragraph:

[0163] To summarize, in an embodiment of the present invention, two hosts, host 1 and host 2, such as host 11 and host 12 in Fig. 3a, coupled to a storage unit for writing and reading information thereto and from, seek concurrent access to a storage unit (such as the storage unit 16, shown in Fig. 3a) through a switch, such as switches 300 and 500 of Figs. 6 and 10a, respectively. This is an [[importantdifference]] important difference with that of prior art systems because while in the prior art, two hosts have access to the storage unit, they cannot concurrently access the same. In the prior art, if a connection between one of the hosts to the storage unit fails for some reason, the other host can continue to access the storage unit. However, switching to the other host, after the detection of a failure, causes a glitch in that the system needs to be reset prior to the other host's communication with the storage unit.

Please replace paragraph [0164] with the following amended paragraph:

[0164] In yet other prior art systems, such as fault-tolerant systems, one host shadows the other host, that is whatever the active host is doing is attempted to be mimicked by the inactive host. This concept is called "heartbeat" indicating a connectivity between the two hosts to the extent both hosts are aware of each other's presence and that the other is operational. That is, one host realizes the failure by the other host in the event this "heartbeat" is no longer detected at which time the host that has performed the detection takes over accessing the storage unit and continues to operate without the other host. Yet [[~~in~~]] such prior art systems require using a dual ported storage unit and can not use a single ported storage unit since the hosts are not capable of accessing the storage unit concurrently as done by the present invention.

Please replace paragraph [0165] with the following amended paragraph:

[0165] Within enterprise systems, there is a great need for the embodiments of the present invention because multiple hosts are required to access a single ported storage unit at the same time. In the present invention, commands are transferred from the hosts to the storage unit concurrently as are other types of [[information ..]] information . The present invention eliminates any glitches caused by switching from an active to an inactive host, as experienced by some prior art systems described hereinabove. In fact, in the present invention, switching between the two hosts is performed in a continuous and smooth fashion.

Please replace paragraph [0167] with the following amended paragraph:

[0167] As shown in Fig. 4, in one of the systems of the prior art, there is a physical layer for one host, another physical layer for the other host and a physical layer for the device or storage unit used by a switch that is coupled between the hosts and the device. None of the other layers are in communication with the hosts and/or device. Through the physical layer, one of the hosts is selected by a multiplexer for communicating [[withthe]] with the device and then the device sends data to that active host. An active host selection circuit decides or selects which host is initially selected along with an initialization circuit. Thus, this prior art switch only needs layer one or the physical layer to communicate, no other layers are needed for communications. However, as noted earlier, one of the problems with such a prior art system is the delay through the switch. Another problem is that only one host can communicate with the device at any given time.

Please replace paragraph [0170] with the following amended paragraph:

[0170] In Fig. 6, concurrent access by two hosts to a device is depicted. Concurrency, as used herein, indicates acceptance of commands, from either of two or more hosts, at any given time including when a device (such as a storage unit) is not in an idle state. Idle state is when the device [[isnot]] processing other commands. Traditionally, concurrency is achieved by multiplexing each host at a given slice of time, or what is commonly referred to as Time Division Multiplexing (TDM). However, this does not work well for storage devices because one may be in the middle of data transfer when suddenly, the transfer is interrupted to service another host due to a new time slice, or slot, occurring, which would be devastating to system performance and may result in lost data.